63-3-5

TM-892/006/00

TECHNICAL MEMORANDUM

(TM Series)

This document was produced by SDC in performance of contract AF 19(628)-1648, Space Systems Division Program, for Space Systems Division, AFSC.

Information for COP Users

Notes of COPII

by

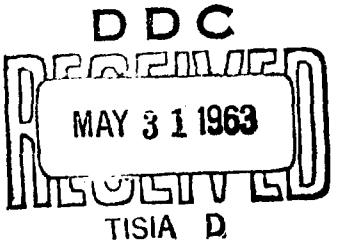
B. R. Pruett

16 April 1963

Approved

J. A. Kneemeyer

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO A
SANTA MONICA
CALIFORNIA



This document is one of a series of TM-892 volumes established to provide information concerning usage and/or operation of the COPII executive system.

Notes on COPII

A. INTRODUCTION

There are a number of restrictions imposed upon the usage of the COPII control program (MTCII), which are not obvious from the existing documentation. This document is designed to clarify these areas and to recommend a few debugging techniques.

B. RESTRICTIONS

1. Element Definition

An element may be loaded from a binary and/or non-binary card deck by MTCII via the CARDS pseudo function. This element is then referred to as a "defined" element of which there are two types; "newly defined", and "re-defined". A "newly defined" element is one which does not exist on the Master Tape. A "re-defined" element is a temporary replacement for an element already on the Master Tape.

The following illustration will be used in the clarification of defined elements:

```
* CARDS      P1
DEFINES      A      ARANGE
   Deck defining A
DEFINES      B      BRANGE
   Deck defining B
DEFINES      C      CRANGE
   Deck defining C
* Function
```

This document is one of a series of TM-892 volumes established to provide information concerning usage and/or operation of the COPII executive system.

Notes on COPII

A. INTRODUCTION

There are a number of restrictions imposed upon the usage of the COPII control program (MTCII), which are not obvious from the existing documentation. This document is designed to clarify these areas and to recommend a few debugging techniques.

B. RESTRICTIONS

1. Element Definition

An element may be loaded from a binary and/or non-binary card deck by MTCII via the CARDS pseudo function. This element is then referred to as a "defined" element of which there are two types; "newly defined", and "re-defined". A "newly defined" element is one which does not exist on the Master Tape. A "re-defined" element is a temporary replacement for an element already on the Master Tape.

The following illustration will be used in the clarification of defined elements:

```
* CARDS        P1
DEFINES        A        ARANGE
   Deck defining A
DEFINES        B        BRANGE
   Deck defining B
DEFINES        C        CRANGE
   Deck defining C
* Function
```

MTCII will initialize its internal tables when the first DEFINES card is encountered. This means that the elements previously in core will not be acknowledged. The first element to be defined will be loaded into core depending upon the P_1 field on the CARDS request. If P_1 is omitted, the first element will be loaded at 10000B; otherwise, P_1 will be used as the starting location (P_1 must be greater than or equal to 10000B). MTCII will load a binary and/or non-binary deck from cards reserving an area equal to the length on the DEFINES card (e.g., ARANGE). At the time a subroutine reference is encountered, when loading an element from cards or prestored cards, a check is made to determine if it has already been assigned a location. If not, an area is reserved for that subroutine equal to the length specified for it in the directory. The starting location for the referenced subroutine is then substituted for its CODES number.

The subroutines referenced by the element being defined are called direct subroutines (D_A). After assigning a starting location to a direct subroutine reference, the environment required by that subroutine is flagged.

An element in the environment of a direct subroutine is called an "indirect" subroutine (e.g., I_A, an indirect subroutine of **A**). After reading one element into core from cards, MTCII will process the next defined element, if any, in the same fashion.

When the first function request is made after defining elements, areas will be reserved for all the indirect subroutines. Then a storage analysis and corrector table are listed, if so specified on the function request. All the direct and indirect elements are loaded from the Master Tape and the requested function is operated.

The restrictions on the usage of defined elements are as follows:

a. Core Requirement

The amount of core required by elements A, B and C, and the "direct" elements of A and B must be less than 60000B cells. The remaining environment of these three elements (D_C, I_A, and I_C) may occupy core up to 76433B. The reason that the former restriction is made is that the code of MTCII, which controls the loading of defined elements, occupies the area of core from 70000B to 76433B, whereas the code which loads elements from the Master Tape is located below 10000B.

b. Storage Analysis

A storage analysis is defined as a list of all elements required to operate the requested function including defined elements and their starting locations. A complete storage analysis will be output only for the first defined function that is requested, e.g.,

```
*    13    A
*    13    B
*    13    C
```

A storage analysis is output when the defined function A is requested, but when B and C are requested, only a storage analysis heading will be output unless B and C were "newly" defined functions. In this case, B and C would be given.

If a function is requested, which references the re-defined elements, those elements will appear in the storage analysis even if the function request does not immediately follow the definitions.

c. MTCII Re-load

If the environment of A, B and C exceeds 70000B, a flag internal to MTCII is set indicating the fact that MTCII must be re-loaded. Assume the following function requests are made after defining A, B and C:

```
*    13    A
*    13    B
```

The function "A" will be processed correctly. After execution of "A", MTCII is re-loaded so that the next function request can be processed. On a re-load, the corrector table is preserved and the SOM, if any, is retained. In re-loading, the previous contents of core is cleared. When the request for the previously defined function "B" is interpreted, one of two actions will be taken; it will either be loaded from the Master Tape, if it exists on the tape, or it will be logged as undefined.

Due to this restriction, it is recommended that elements be defined at the time at which they are needed.

d. DEFINES Length

When a function or element is re-defined, the new length from the DEFINES card is not stored in the directory. If SYMDUMP is used to dump that element, the length from the directory is used. If the length of the element on the Master Tape is less than the new element, part of the new element will not be listed.

To provide a means of temporarily changing the length of a program in the directory, the REDEFINE pseudo is available and is described in TM-745, MTCII Program Description.

e. Absolute Elements

An absolute element must observe the restrictions imposed on relocatable elements, in that it must fit into the area between 10000B and 70000B, if it is loaded via CARDS.

f. Subroutine Reference

A defined element cannot reference a subroutine which is in the CODES table and not on the Master Tape, even though the subroutine has been loaded into core via CARDS. At the present time, intercommunication between "newly defined" elements whose names appear in the CODES table is not possible. This capability will be implemented in Mod AG of MTCII.

g. Successive Requests for Definitions

Successive requests for a defined function are possible only if the defined function and its environment do not exceed location 70000B. A storage analysis will be given on the first request but only the "newly defined" elements will be displayed in the following analyses. If the 70000B limit is exceeded, the second request for the function will be either undefined, or the function will be loaded from the Master Tape. In the latter case, an analysis of the element from the Master Tape will be output.

h. Definition in Special Operating Mode

1) MTCII must already be in the Special Operating Mode (SOM) prior to defining a function which utilizes the SOM mode.

2) The function specified in SPETAB, which is operated by MTCII when the SOM mode is initialized, cannot be defined. It must be on the Master Tape.

3) To remain in the SOM mode when defining elements, the P_1 field on the CARDS request must be equal either to RCLANK or to an address which is greater than the reserved area that starts at 10000B. If RCLANK is used, MTCII will load the first defined element starting at the first location following the reserved area. In either case, core will be initialized except for the reserved areas.

4) To request a function which has previously been defined, an N_3 field must not precede the function name.

	INCORRECT		CORRECT
	* RCLANK	A	* A
or	* N_3	A	* A

An N_3 control parameter initializes the core allocation table, which will delete the defined elements.

2. Usage of N_3

N_3 is a control parameter which may precede the name of the function being requested. N_3 provides a method by which the starting location for a function may be specified. N_3 may be equal to RCLANK or greater than or equal to 10000B.

3. Successive Requests for Functions

If a function is requested more than once in succession, one of the following events will occur, depending upon the size of area required by the function and its environment.

a) A function and its environment which exceeds 60000B cells in length will result in the re-load of MTCII after the operation of the first request. The second or more requests for the function will result in the function and its environment being loaded for each request.

b) A function and its environment, which does not exceed 60000B cells in length, will not be re-loaded for each request.

4. Usage of N_3 in SOM Mode

To request a function which does not utilize the SOM mode, yet retain the mode, the N_3 control parameter must be specified on the function request, e.g.,

```
* RCLANK DUMP 3 10B 100B
```

where N_3 = RCLANK. MTCII will automatically equate the location counter to the first address available following the SOM reserved area whenever N_3 is equal to RCLANK. This usage of N_3 eliminates the need for knowing the length of the reserved area. In addition, the core allocation table is initialized so that MTCII is not aware of previous elements in core. For this reason, SYMDUMP cannot be used in the SOM mode since it depends upon the core allocation table.

5. Core Allocation

When a normal function request is made, that is, the N_3 control parameter is omitted, MTCII will allocate core in the following manner:

a) The environment of the function will be assigned locations in the reverse order of the CODES table. The element of the environment, which appears last in the CODES table, will be assigned the first available location (10000B, if core has been initialized or the function and its environment requires more room than is available).

b) The function will be assigned the first available address following its environment. A flag is set so that after the operation of the function it will be wiped out.

c) When the next function is requested, MTCII will try to fit its environment into core immediately following the environment of the

first function (the last function will be wiped out), providing the second function is not DUMP, SYMDUMP, or the same as the previous function. A special case is made for SYMDUMP and DUMP to provide for debugging the previous function. Elements already in core will not be re-loaded.

If the second function is for DUMP or SYMDUMP, the first function will be retained and the environment of DUMP or SYMDUMP will follow the first function, and the second function will precede its environment. If there is not enough room for the second function between the first available address and 70000B, core is initialized and steps a and b will be performed.

d) Core Allocation Example

The following example illustrates the core allocation process:

Element requirements are:

VMT requires OUTPUT and UNPACK
COS requires UNPACK, FLOATBIN and SIN
DUMP requires CORE

The function request for VMT (* VMT) will result in core being allocated as follows:

10000	OUTPUT
11322	UNPACK
11334	VMT

2. Iterations of Defined Elements

It may be desirable in the checkout of an element to define it and call it whenever it is desired to operate it. In the COPII system, there is no guarantee that a defined element will remain in core if other functions are operated between requests for this defined element; therefore, it is necessary to define the function each time it is to be requested. To eliminate the need for numerous binary decks of the function, the definition of the function can be prestored and called when needed by the TAPERD pseudo. There are two steps to this process.

a) Prestore the defined function as follows:

```
* CARDS
DEFINES    F1     RANGE
   Deck of F1
* X   13  F1
```

The request for F_1 is made at this point to merely load the environment of the function, to obtain a storage analysis, and to terminate the defines mode. The operation of F_1 is not actuated by this request since the "Do not Operate" parameter (X) is used. Instead, the function is requested again in the normal card setup (see step b) so that variable parameters may be specified. Note that this scheme will work only for functions which do not exceed location 70000B.

b) To operate the defined function from the prestored tape, the following sequence must be executed:

```
* REWIND  UNIT
* TAPERD  UNIT
* F1   P1   P2....Pp, M1....Mm
```

This sequence results in rewinding the prestored tape, reading the defined program from this tape, loading its environment from the Master Tape, and executing the function.

c) The following example illustrates the usage of this mode of operation.

```
* IRT          2     9     10
* WNRT
* REWIND       U
* TAPERD       U
* F1  P1.....Pp
* RUNNUM
* KINJFUN
* REWIND       U
* TAPERD       U
* F1  P1.....Pp,.....
```

d) More than one deck can be defined following the CARDS card, as well as CORRECT blocks, i.e.,

```
* CARDS
CORRECT     F1
     Correctors
CORRECT     F2
     Correctors
DEFINES     F3     RANGE
     Definition of F3
DEFINES     F4     RANGE
     Definition of F4
*   13   X  F3
```

16 April 1963 TM-892/006/00

DISTRIBUTION LIST

External

Space Systems Division
(Contracting Agency)
Major C. R. Bond (SSOCD)

6594th Aerospace Test Wing
(Contracting Agency)
Col. A. W. Dill (TWRD)
Lt. Col. M. S. McDowell (TWRU) (4)
TWACS

PIR-E1 (Lockheed)
N. N. Epstein
C. H. Finnie
H. F. Grover
H. R. Miller
W. E. Moorman (5)
461 Program Office
698BK Program Office

PIR-E2 (Philco)
J. A. Bean
J. A. Isaacs
R. Morrison
S. M. Stanley

PIR-E3 (LFE)
D. F. Criley
K. B. Williams (5)

PIR-E8 (Mellonics)
F. Druding

PIR-E5 (Aerospace)
F. M. Adair
R. V. Bigelow
R. D. Brandsberg
L. H. Garcia
G. J. Hansen
C. S. Hoff
L. J. Kreisberg
T. R. Parkin
E. E. Retzlaff
H. M. Reynolds
D. Saadeh
R. G. Stephenson
V. White

PIR-E7 (STL)
A. J. Carlson
R. L. Mills

PIR-E4 (GE-Sunnyvale)
D. Alexander
J. Farrentine
N. Kirby

PIR-E4 (GE-Box 8555)
J. S. Brainard
H. G. Klose
J. D. Selby

PIR-E4- (GE-3198 Chestnut)
J. F. Butler
H. D. Gilman

PIR-E4 (GE-Bethesda)
W. L. Massey

PIR-E4 (GE-Box 8661)
J. D. Rogers

TOTSCHEK, R. A.	24120	WINSOR, M. E.	22156
VORHAUS, A. H.	24076A	WINTER, J. E.	24117
WAGNER, I. T.	24093	WISE, R. C.	22085
WARSHAWSKY, S. B.	24097	WONG, J. P.	SUNNYVALE
WEST, G. D.	SUNNYVALE	ZACHTE, S. A.	24086
WEST, G. P.	22116A	ZUBRIS, C. J.	24075
WILSON, G. D.	24124		
BUSCH, R. E.	22088B		
MARSHALL, R. D.	22160		

DISTRIBUTION LIST
Internal

AFCPL	(5)	KEDDY, J. R.	24105
ALLFREE, D.	24083	KEY, C. D.	23013
ALPERIN, N. I.	22153	KEYES, R. A.	24073
ARMSTRONG, E.	24123	KINKEAD, R. L.	22093
BERNARDS, R. M.	SUNNYVALE	KNEEMEYER, J. A.	22088A
BIGGAR, D.	24118A	KNIGHT, R. D.	22119
BILEK, R. W.	23007	KOLBO, L. A.	22155
BLACK, H.	14039	KOSTINER, M.	14056B
BRENTON, L. R.	24103B	KRALIAN, R. P.	14039
BURKE, B. E.	24086	KRISTENSEN, K.	SUNNYVALE
BURKE, R. F.	22158	LACHAPELLE, F.	22093
CHAMPAIGN, M. E.	22152	LAUGHLIN, J. L.	24073
CHIODINI, C. M.	24091	LAVINE, J.	24093
CIACCIA, B. G.	24082A	LITTLE, J. L.	24088B
CLINE, B. J.	24127	LONG, F.	22156
COGLEY, J. L.	22156	MADRID, G. A.	22081
CONGER, L.	24088A	MAHON, G. A.	24089
COOLEY, P. R.	24081	MARIONI, J. D.	24076B
COURT, T. D.	24086B	MARTIN, W. P.	24127B
CRUM, D. W.	24105	MCKEOWN, J.	23013
DANT, G. B.	24086B	MICHAELSON, S. A.	14039
DECUIR, L. E.	24053A	MILANESE, J. J.	22155
DERANGO, W. C.	24082B	MUNSON, J. B.	22087A
DEXTER, G. W.	25016	MYERS, G. L.	14056A
DISSE, R. J.	23014	NELSON, P. A.	24075
DOBBS, G. H.	22116B	NG, J.	22077
DOBRUSKY, W. B.	24065A	NGOU, L.	24127
ELLIS, R. C.	22131A	PADGETT, L. A.	24110A
EMIGH, G. A.	14039	PATIN, O. E.	SUNNYVALE
ERICKSEN, S. R.	22113	POLK, T. W.	24113
FELKINS, J.	24097	PRUETT, B. R.	22084
FOSTER, G. A.	14039	RAYBIN, M.	14039
FRANKS, M. A.	24122	REILLY, D. F.	24121
FREY, C. R.	22078	REMSTAD, C. L.	25026
FRIEDEN, H. J.	22082	RUSSELL, R. S.	14054
GARDNER, S. A.	25026	SCHOLZ, J. W.	14039
GREENWALD, I. D.	22094A	SCOTT, R. J.	24110
GRIFFITH, E. L.	22081	SEACAT, C. M.	SUNNYVALE
HAAKE, J. W.	22153	SEIDEN, H. R.	22126B
HARRIS, E. D.	24081	SHAPIRO, R. S.	24110B
HENLEY, D. E.	22094B	SKELTON, R. H.	22148
HILL, C. L.	22101	SOLOMON, J.	22076
HILLHOUSE, J.	22078	SPEER, N. J.	24086A
HOLMES, M. A.	24103	STONE, E. S.	24058B
HOLZMAN, H. J.	24065B	SWEENEY, M. J.	25026
HOUGHTON, W. H.	24103B	TABER, W. E.	22101
HOYT, R. L.	14039	TENNANT, T. C.	27029
IMFL, L. E.	14039	TESTERMAN, W. D.	14039
KASTAMA, P. T.	22076	THOMPSON, J. W.	24088
KAYSER, F. M.	24109	THORNTON, R. L.	14050

UNCLASSIFIED

System Development Corporation,
Santa Monica, California
INFORMATION FOR COP USERS NOTES
ON COPII.
Scientific rept., TM-892/006/00,
by B. R. Pruett. 16 April 1963, 12p.
(Contract AF 19(628)-1648, Space
Systems Division Program, for Space
Systems Division, AFSC)
Unclassified report

DESCRIPTORS: Programming (Computers).
Satellite Networks.

UNCLASSIFIED

UNCLASSIFIED

Clarifies restrictions imposed
upon the usage of the COPII
control program (MTCII),
and recommends a few debugging
techniques.

UNCLASSIFIED